Differentiable Kernel Evolution

Yu Liu[1,3*] Jihao Liu[2*] Ailing Zeng[3] Xiaogang Wang[1,3]
{yuliu@ee, xgwang@ee, alzeng@cse}.cuhk.edu.hk liujihao@sensetime.com
[1]CUHK-SenseTime Joint Laboratory
[2]SenseTime Research [3]The Chinese University of Hong Kong

Abstract

This paper proposes a differentiable kernel evolution (DKE) algorithm to find a better layer-operator for the convolutional neural network. Unlike most of the other neural architecture searching (NAS) technologies, we consider the searching space in a fundamental scope: kernel space, which encodes the assembly of basic multiply-accumulate (MAC) operations into a conv-kernel. We first deduce a strict form of the generalized convolutional operator by some necessary constraints and construct a continuous searching space for its extra freedom-of-degree, namely, the connection of each MAC. Then a novel unsupervised greedy evolution algorithm called gradient agreement guided searching (GAGS) is proposed to learn the optimal location for each MAC in the spatially continuous searching space. We leverage DKE on multiple kinds of tasks such as object classification, face/object detection, large-scale fine-grained and recognition, with various kinds of backbone architecture. Not to mention the consistent performance gain, we found the proposed DKE can further act as an auto-dilated operator, which makes it easy to boost the performance of miniaturized neural networks in multiple tasks.

1. Introduction

Recently several works [2, 7, 10, 32] notice that a well-designed / learnable kernel shape can boost the performance of a convolutional neural network (CNN) on some specific tasks without modifying the macroscopical architecture. For a conventional kernel w with shape of $k \times k \times c_{in}$ in the discrete spatial space, some of them [7] directly dilate the pixels of a convolution kernel by 0, increasing the kernel shape to $nk \times nk \times c_{in}$, so that it can hold a larger receptive field. Some of the others [2, 32] claim that learning a data-sensitive offset $O, O \in \mathbb{R}^{W \times H \times k \times k \times 2}$ to w is a good choice, which enables the network to find an *ad-hoc* receptive field based on what it is receipting at each location. However [10] holds the point that a fixed and shared offset $O, O \in \mathbb{R}^{2 \times k \times k}$ is good for generalization. Since all of these works based on different environments and tasks and thus have different motivations, they may lose generalization in some other tasks. For example, DCNs [2, 32] mainly focus on object detection tasks, where the context of the input image varies, so a spatial attention mechanism is needed and it is natural to predict specific offsets for w at different locations. However for tasks with fixed input size and well-aligned input data like face classification and fine-grain recognition, it is not preferred to introduce extra uncertainty like data-dependent kernel shape.

To find a generalized direction for the kernel design, this work does not focus on one specific task but the convolution operation itself. We first formulate a generalized operator consisting of MAC operation between the input data and kernel. The generalized form covers all the related works mentioned above. After that, without any prejudice, we construct two necessary constraints over it to maintain the two appealing properties, namely *equivalence to translation* and *independence to data*. Given the constraints, the final form will be degenerated to a generalized convolution operator, with the freedom of MAC location in each kernel and each channel k. We will refer to the locations of MACs as a set V_k, which consists of "valid coordinate" $\mathbf{y}$ in a kernel. Note that the kernel size at each location is denoted as k'. Then the main contribution of this work is to propose a differentiable kernel evolution (DKE) algorithm to search optimal locations for the $k' \times c_{in}$ MACs in a spatial continuous space.

Searching the continuous coordinate $\mathbf{y}$ with discrete input data sources is difficult. In this work, a greedy searching algorithm is constructed. Similar to the other works, DKE visits the data value at a continuous point by interpolation. However, we found the commonly used *nearest* and *bilinear* interpolations naturally harm the searching process. Since the gradient direction of the interpolative function controls the evolution direction during each searching step (see Sec. 4 for details), it should be in the same direction to the *ad hoc* optima in each iteration. Un-

*They contributed equally to this work

fortunately, neither *nearest* nor *bilinear* naturally holds this property. This is to say, a new competent interpolative function is needed.

To achieve that, we construct three basic rules, namely *continuity*, *monotonicity* and *gradient agreement*, for the optimal interpolative function. We theoretically demonstrate their necessary for the greedy search. Then a novel interpolative kernel is proposed and satisfy the mentioned constraints. In this way, the stability can be guaranteed.

This work takes significant effort to find a good interpolative function. The motivation of that is to theoretically find a way to embed the greedy searching process into the backpropagation of the network. This design enables the whole system to be optimized together without any iterative training, such as some reinforcement learning methods [33, 34, 30] do. Despite the intricate supporting theories, thorough experiments show the generalization ability and advancement of DKE. We embed DKE-conv to multiple network structure like AlexNet [14], ResNet [6], RetinaNet [18] and FPN [17], and then verify them in multiple challenging tasks, such as Cifar [13], ImageNet [3], 1-million face recognition [12], face detection [9, 31] and COCO [19] object detection.

Some insights and interesting points of this paper can be concluded as follows:

+ We theoretically deduce and demonstrate a most generalized form of MAC-based operator for the convolutional neural network;
+ A novel differentiable greedy searching algorithm is proposed based on the designed searching space;
+ We theoretically demonstrate the gradient agreement is significant for the differentiable searching, and by that we further deduce a novel interpolation algorithm for continuous space approximation;
+ The generalized form of the DKE ensures its ability of generalization. Consistent performance gain can be provided on most modern tasks with arbitrary backbones.
+ Experiments also show that DKE can act as an auto-dilated operator for shallow neural networks, which boosts the performance of light-weight model by a significant margin.

2. Related Work

2.1. Irregular Kernel

Most targets in tasks like classification, detection and recognition are structured with various shapes. However, the conventional convolution only holds a rectangular receptive field. Several recent works noticed this problem, and proposed different methods to figure it out. STN [8] is a good start to learn spatial transformation from data, which is optimized with SGD in an end-to-end manner. It learns an affine transformation via backpropagation, which is further applied to the feature map, trying to get an invariant expression of the feature. Such a global transformation is inefficient and difficult to learn, which cannot be applied to large scale datasets like ImageNet [3]. Instead, the proposed DKE embeds a more efficient differentiable transformation into each convolutional kernel. On the other side, AUC [10] and DCN [2, 32] tries to learn spatial offsets to model transformation. We will study the essential relationship and difference over these and our algorithm in Sec. 4.

2.2. Neural Architecture Searching (NAS)

In general, there are mainly three schools for the NAS, which tries to search for a good CNN architecture automatically. The first is based on reinforcement learning (RL). [33] is the first to use to search for NN architecture. Further, [34] proposed a new searching space named NASNet search space. Besides RL-based method, [26] tries to leverage the evolution algorithm to search neural network architecture more efficiently. In addition, [20] models NAS in a differentiable way. They add a gate to each operation and the network is optimized via backpropagation end-to-end. However, even though our work also proposes a differentiable way for network evolution, it considers the searching space in a much fundamental kernel scope, and the searching algorithm of them are quite different.

3. Differentiable Kernel Evolution

Our goal is to find a generalized convolution-like operation that only contains multiply-accumulate (MAC). This new operation should still be competent for any CNN based tasks such as classification and object detection, so it should obey the *equivalence constraint* and *independent constraint*. To explore how far we can break through the conventional convolution layer, in Sec. 3.1 we first formulate an assembly of MAC operations as Eq. 1, namely generalized convolution. Then we degenerate it by the two constraints. We will show the final form Eq. 9 can be deemed as a generalized convolution with the additional parameter set V_k. Then Sec. 3.2 will introduce the details of searching space and searching algorithm for finding an optimal V_k.

3.1. Generalized Convolution

Formulation Formally, denote the image/feature map coordinate by $\mathbf{x}, \mathbf{y}, \mathbf{t} \in \mathbb{Z}^2$, and the pixel values at each coordinate by a function $f : \mathbb{Z}^2 \mapsto \mathbb{R}^K$, where K is the number of input channels. The ideal operator is based on MAC, thus it follows the form as $\sum f \cdot w$ where w is the parameter of the operator, and can also be formulated as a function $w : \mathbb{Z}^2 \mapsto \mathbb{R}^K$. Based on these denotations, a generalized

MAC operation can be represented as:

$$[f * w](\mathbf{x}) = \sum_{\mathbf{y}_f, \mathbf{y}_w \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(\mathbf{y}_f) w_k^{\mathbf{x}}(\mathbf{y}_w(\mathbf{x})) + b_k. \quad (1)$$

Note that: **1)** for each output location $\mathbf{x}$, we have a set of kernel $\{w^{\mathbf{x}}\}$ and an offset of kernel coordination $\mathbf{y}_w(\mathbf{x})$; **2)** the value of f and w are equal to zero outside the input feature map and operator valid region, respectively. We call the non-zero region *activated field*, donated as a coordinate set V_k for the k-th channel. For example, for a conventional 3×3 convolution kernel with dilation 2, the $w_k^{\mathbf{x}}$ will be zero except at location belongs to $V_k \equiv \{(-2,-2),(-2,0),(-2,2),...,(2,2)\}$ for all $\mathbf{x}$ and k, and $\mathbf{y}_w(\mathbf{x}) \equiv \mathbf{y}_f - \mathbf{x}$. But for deformable convolution [2] the non-zero area of $\{w_k^{\mathbf{x}}\}$ varies for different $\mathbf{x}$. Then the MAC is operated between f_k and $\{w_k^{\mathbf{x}}\}$.

Keep these in mind, we now formulate two constraints on 1:

Constraint 1. ***(Equivalence constraint)*** *The ideal operator w should be equivalent to translation.*

Conventional convolution layer holds the appealing property named equivalence to translation, *i.e.*, the result of translating the input is equivalent to translating the output feature map, which makes it naturally support sliding window-based tasks such as object detection, segmentation and key-point detection. Now consider the ideal operation, let $L_{\mathbf{t}}$ denote the coordinate translation $\mathbf{x} \rightarrow \mathbf{x} + \mathbf{t}$ on the feature map, *i.e.*,

$$[L_{\mathbf{t}} f](\mathbf{x}) = f(\mathbf{x} - \mathbf{t}). \quad (2)$$

$$[[L_{\mathbf{t}} f] * w](\mathbf{x}) = [L_{\mathbf{t}}[f * w]](\mathbf{x}) \quad (3)$$

For the left item, we have

$$\begin{aligned}
[[L_{\mathbf{t}} f] * w](\mathbf{x}) &= \sum_{\mathbf{y}_f, \mathbf{y}_w \in \mathbb{Z}^2} \sum_{k=1}^{K} [L_{\mathbf{t}} f]_k(\mathbf{y}_f) w_k^{\mathbf{x}}(\mathbf{y}_w(\mathbf{x})) \\
&= \sum_{\mathbf{y}_f, \mathbf{y}_w \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(\mathbf{y}_f - \mathbf{t}) w_k^{\mathbf{x}}(\mathbf{y}_w(\mathbf{x})) \\
&= \sum_{\mathbf{y}_f, \mathbf{y}_w \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(\mathbf{y}_f) w_k^{\mathbf{x}}(\mathbf{y}_w(\mathbf{x}) + \mathbf{t}),
\end{aligned} \quad (4)$$

and for the right item,

$$[L_{\mathbf{t}}[f * w]](\mathbf{x}) = \sum_{\mathbf{y}_f, \mathbf{y}_w \in \mathbb{Z}^2} \sum_{k} f_k(\mathbf{y}_f) w_k^{\mathbf{x}}(\mathbf{y}_w(\mathbf{x} - \mathbf{t})). \quad (5)$$

By aggregation of Eq. 4 and Eq. 5, followed by central normalization for the kernel, we can easily get the final form of equivalence constraint:

$$\mathbf{y}_w(\mathbf{x}) = \mathbf{y}_f - \mathbf{x}. \quad (6)$$

Constraint 2. ***(Independent constraint)*** *The weight of operator $w_k^{\mathbf{x}}$ and its activated field should be independent to the location of input data $\mathbf{x}$.*

While some works like deformable convolution [2] predicts different *activated field* V_k of $w_k^{\mathbf{x}}$ for different $\mathbf{x}$ and gain considerable improvement for some specific tasks like object detection, they usually lose generalization in other tasks especially when the input image is well aligned (see experiments in Sec. 5.2). Without loss of generality, here we consider a data-independent kernel:

$$\forall \mathbf{x}_i, \mathbf{x}_j, w_k^{\mathbf{x}_i} = w_k^{\mathbf{x}_j} \quad (7)$$

Substituting Eq. 6 and Eq. 7 into Eq. 1, we can get the formulation of the generalized operator that obey the constraints above:

$$[f * w](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \sum_{k} f_k(\mathbf{y}) w_k(\mathbf{y} - \mathbf{x}) + b_k. \quad (8)$$

It can be simplified as follow for comprehensible:

$$[f * w](\mathbf{x}) = \sum_{k} \sum_{\mathbf{y} \in V_k} f_k(\mathbf{x} + \mathbf{y}) w_k(\mathbf{y}) + b_k, \quad (9)$$

We denote the $s = card(V_k)$ indicates the coordinate number of w, so w has the shape $s \times K$, which is more flexible compared with conventional kernel shape $s_w \times s_h \times K$. Practically V_k act as a filter that selecting a certain set of locations for input map and kernel to perform the MAC as shown in Fig. 1, which is equivalent to selecting the connection between the kernel and input. Namely, for each output location $\mathbf{x}$, we find a set of valid offset $\mathbf{y} \in V_k$, which build up a set of connections $\{\mathbf{x} + \mathbf{y} \leftrightarrow \mathbf{y}\}$ for input f_k and kernel weight w_k to finish one MAC operation $f_k(\mathbf{x}+\mathbf{y}) \cdot w_k(\mathbf{y}) + ...$ Obviously an optimal V_k is needed for a good performance. We will introduce how to search it in the next section.

3.2. Searching Space Setup

Without loss of generality, we consider $\mathbf{y} \in V_k \subset \mathbb{R}^2$ rather than $\mathbb{N}^2$ and $w : \mathbb{R}^2 \mapsto \mathbb{R}^K$ for the sake of the derivability. A well-designed interpolation (introduced in Sec. 3.3) will be used to estimate the value of f and w at non-integer coordinates. At the beginning, V_k is initialized by Gaussian sampling. Namely, at the beginning, the algorithm samples s locations $\mathbf{y}_{1...s} \in V_k$ from the standard Gaussian distribution $\mathcal{G}(\mu, \sigma)$. Obviously the conventional

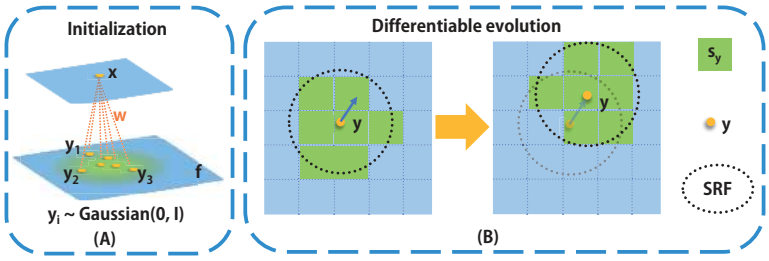


Figure 1. Illustration of the differentiable kernel evolution. **(A)** Initialization of DKE. The blue square indicates the input (large) and output (small) feature map. **(B)** The searching process in one iteration. Best view in color.

convolution, whose kernel has a fixed shape $s_w \times s_h$, is a special case of this design.

During one training iteration, every MAC at offset $\mathbf{y}$ takes a circle searching-receptive-field (SRF) with radius r, finds the best evolutionary direction $\mathbf{\Delta y}$,and update $\mathbf{y}$ to $\mathbf{y} + lr\mathbf{\Delta y}$ greedily. However, visiting and evaluating all continuous locations in SRF are time-consuming. Next, we will introduce a novel differentiable searching algorithm named gradient agreement guided searching (GAGS) to reduces the searching latency while outperforming most of the state-of-the-art differentiable searching method.

3.3. Gradient Agreement Guided Searching

We introduce a greedy searching algorithm named gradient agreement guided searching to search the V_k and can be embedded into the backpropagation by a meticulously designed interpolation function.

As shown in Fig. 1, each valid offset $\mathbf{y} \in V_k$ maintains a local searching area $S_\mathbf{y}$ centred on $\mathbf{y}$ with searching radius r in each iteration, namely, $\forall \mathbf{s_y} \in S_\mathbf{y}, \mathbf{s_y} \in \mathbb{N}^2$ and $\|\mathbf{s_y} - \mathbf{y}\|_2 <= r$. Keeping the consistent meaning of $\mathbf{x}$ as above, $\mathbf{x} + \mathbf{y} \in \mathbb{R}^2$ may not be an integer coordinate, so we need to interpolate an applicable value for $f_k(\mathbf{x} + \mathbf{y})$ in Eq. 9.

$$f_k(\mathbf{x} + \mathbf{y}) = \sum_{\mathbf{s_y} \in S_\mathbf{y}} \mathcal{G}(\mathbf{s_y}, \mathbf{y})^1 * f_k(\mathbf{x} + \mathbf{s_y}) \tag{10}$$

The interpolation function $\mathcal{G}$ return the weights of the $\mathbf{s_y}$ that contributes to the continuous location $\mathbf{y}$, which should strictly obey the following three rules:

Constraint 3. ***Continuity:*** *$\mathcal{G}(\mathbf{s_y}, \mathbf{y})$ should be differentiable almost everywhere w.r.t. $\mathbf{y} \in \mathbb{R}^2$.*

Constraint 4. ***Monotonicity:*** *$\mathcal{G}(\mathbf{s_y}, \mathbf{y})$ should be monotonically non-increasing w.r.t. $|\mathbf{s_y} - \mathbf{y}|$.*

Constraint 5. ***Gradient Agreement:*** *the partial derivative $\frac{\partial \mathcal{G}}{\partial \mathbf{y}}$ should be $C \cdot (\mathbf{s_y} - \mathbf{y})$ where $C \in \mathbb{R}^+$, i.e. $\frac{\partial \mathcal{G}(\mathbf{s_y}, \mathbf{y})}{\partial \mathbf{y}}$ and $(\mathbf{s_y} - \mathbf{y})$ should point in the same direction.*

[1]Originally it should be $\mathcal{G}(\mathbf{x} + \mathbf{s_y}, \mathbf{x} + \mathbf{y})$, but this equation leaves $\mathbf{x}$ based on the shift-invariance of the interpolation function: $\mathcal{G}(\mathbf{x} + \mathbf{s_y}, \mathbf{x} + \mathbf{y}) = \mathcal{G}(\mathbf{s_y}, \mathbf{y})$.

Note that most common interpolation functions like *bilinear* and *nearest neighbor* not satisfy these constraints due to the lack of continuity and gradient agreement, respectively. Details will be shown in Sec. 4.

The **continuity** and **monotonicity** rules are nature since the interpolation space of $\mathbf{y}$ should be derivable during backpropagation, and the closer the reference point is to the interpolation point, the higher, at least not lower, the coefficient of the reference point.

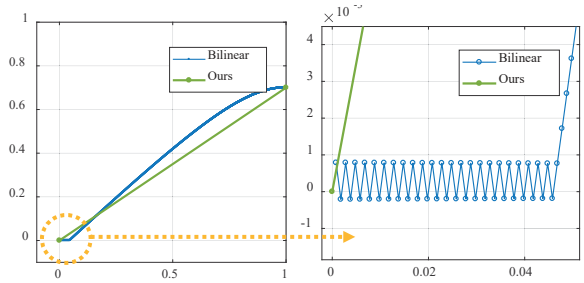


Figure 2. Simulated searching paths of $\mathbf{y}$ by gradient descent in the discrete searching space interpolated by different kernels. The initial coordinate of $\mathbf{y}$ is set to [1,0.7] and the optimal location is [0,0]. Bilinear interpolation leads to an unsteadiness due to the lack of gradient agreement.

Now we deduce why **gradient agreement** is necessary for the *greedy* searching process:

Proof. Denoting the output of the layer (summation of the MACs) and the loss function as $\mathbf{o}$ and $\mathcal{L}$ respectively, then from Eq. 9 and 10, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{y}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{o}} \cdot \frac{\partial \Big(\sum_{k,\mathbf{y},\mathbf{s_y}} [\mathcal{G}(\mathbf{s_y}, \mathbf{y}) * f_k(\mathbf{x} + \mathbf{s_y})] w_k(\mathbf{y}) + b_k \Big)}{\partial \mathbf{y}} \\ &= \frac{\partial \mathcal{L}}{\partial \mathbf{o}} \cdot \sum_k w_k(\mathbf{y}) \sum_{\mathbf{y},\mathbf{s_y}} f_k(\mathbf{x} + \mathbf{s_y}) * \frac{\partial \mathcal{G}(\mathbf{s_y}, \mathbf{y})}{\partial \mathbf{y}}.\end{aligned} \tag{11}$$

In each addend $\frac{\partial \mathcal{L}}{\partial \mathbf{o}} \cdot w_k(\mathbf{y}) f_k(\mathbf{x} + \mathbf{s_y}) * \frac{\partial \mathcal{G}(\mathbf{s_y}, \mathbf{y})}{\partial \mathbf{y}}$, the positive or negative value of the former scalar $\frac{\partial \mathcal{L}}{\partial \mathbf{o}} \cdot w_k(\mathbf{y}) f_k(\mathbf{x} +$

$\mathbf{s_y}$) implies the MAC ($w_k^{\mathbf{y}\top} \cdot f_k^{\mathbf{x}+\mathbf{s_y}}$) at the integer location $\mathbf{x}+\mathbf{s_y}$ has positive or negative contribution to the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{o}}$. Given the monotonicity of $\mathcal{G}$ and based on the greedy strategy, the local $\mathcal{L}$ can be minimized when $\mathbf{y}_{greed}$ reaches $\mathbf{s_y}$ or out of the searching region for positive and negative cases, respectively. So the greedy $\Delta\mathbf{y} = \mathbf{y}_{greed} - \mathbf{y}$ equals to:

$$\Delta\mathbf{y} = \begin{cases} \mathbf{s_y} - \mathbf{y} & , \frac{\partial \mathcal{L}}{\partial \mathbf{o}} \cdot w_k^{\mathbf{y}\top} \cdot f_k^{\mathbf{x}+\mathbf{s_y}} \geqslant 0 \\ -\gamma(\mathbf{s_y} - \mathbf{y}), \gamma \in \mathbb{R}^+ & , \text{else}, \end{cases} \tag{12}$$

where $\gamma >= \frac{r}{\|\mathbf{s_y}-\mathbf{y}\|_2} - 1$. Let $\Delta\mathbf{y} \propto \frac{\partial \mathcal{L}}{\partial \mathbf{y}}$ and together with Eq. 11 & 12, finally we have

$$\frac{\partial \mathcal{G}(\mathbf{s_y}, \mathbf{y})}{\partial \mathbf{y}} = C \cdot (\mathbf{s_y} - \mathbf{y}), C \in \mathbb{R}^+. \tag{13}$$

■

Based on the rules above, we construct the family of negative-exponential function to be the basic form of interpolation function:

$$\mathcal{G}(\mathbf{s_y}, \mathbf{y}) = \frac{1}{\mathcal{C}} * \exp(-\alpha \|\mathbf{s_y} - \mathbf{y}\|_2^2), \tag{14}$$

where $\mathcal{C}$ is the normalizing constant. It's partial derivative,

$$\frac{\partial \mathcal{G}}{\partial \mathbf{y}} = 2\alpha \mathcal{G}(\mathbf{s_y}, \mathbf{y}) * (\mathbf{s_y} - \mathbf{y}), \tag{15}$$

not only hold the three good properties mentioned above, but has negligible computational workload — 4 multiply–accumulate operations for each point pair compared with 10+ in bilinear.

Fig. 2 shows the searching path in a simplified simulation, considering one $\mathbf{s_y}$ occurs in the SRF at location $\mathbf{0}$ and $\mathbf{y}$ is initialized at [1, 0.7], and the scalar part in Eq. 11 is positive. In this case, the optimal solution for $\mathbf{y}$ is $\mathbf{0}$. Based on the searching space interpolated by the proposed kernel, the $\mathbf{y}$ fast converges to the optimal point due to the good property of *gradient agreement*, while for that interpolated by bilinear does not hold the *gradient agreement*, the $\mathbf{y}$ moves in a curve path and even hangs back when close to the optima. Detail explanation will be introduced in Sec. 4.

In this way, the valid location $\mathbf{y} \in V_k$ can be searched together with the backpropagation.

Since the operator searched by GAGS is data-independent and location-independent, the V_k is shared over all spatial coordinates. To adapt the learned kernel in different cases, here we add a light weighted self-attention operator on the generalized convolution. First, an attention score $\mathbf{a}$ will be produced by a simple 1×1 convolution for all $\mathbf{y}$ at each coordinate: $\mathbf{a_x} = [gf](\mathbf{x}+\mathbf{y})) \in \mathbb{R}^{k'}$. After that, the attention score will be multiplied in each corresponding MAC. That is,

$$[f * w](\mathbf{x}) = \sum_k \sum_{\mathbf{y} \in V_k} \mathbf{a_x} f_k(\mathbf{x}+\mathbf{y}) w_k(\mathbf{y}) + b_k. \tag{16}$$

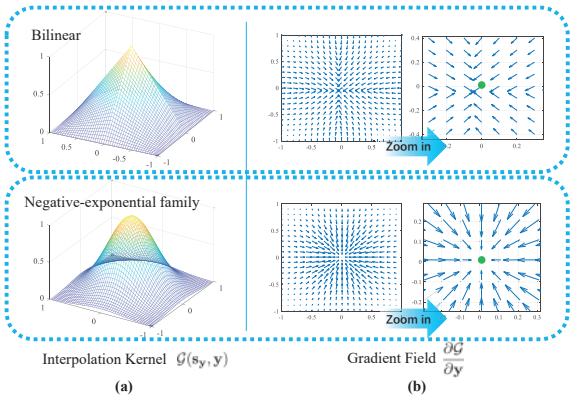


Figure 3. Kernel space and gradient field of different interpolation methods. Note that the green point indicate the reference point $\mathbf{s_y}$ and is also the optima of $\mathbf{y}$ in this case. The lack of gradient agreement occurs almost everywhere in the bilinear's gradient field.

3.4. Details and Review

Considering most of the related works experiment on the kernel with the commonly used shape $3 \times 3 \times K$, based on the principle of a fair comparison, we set $k' = 9$, *i.e.* $card(V_k) = 9$ in all of our experiments. In the aspect of searching space, the radius of SRF is set to 2 ($r = 2$), and the hyper-parameter α is set to 2 for all experiments, but empirically the convergence and performance are not so sensitive to α when $\alpha \in [1e0, 1e1]$.

Given a network architecture, we first initialize the V_k in each kernel by sampling $\mathbf{y} \sim \texttt{Gaussian}(\mathbf{0}, \mathbf{I})$ i.i.d., then we train the network on an arbitrary task, and the optimal value of $\mathbf{y}$ will be automatically searched along with the backpropagation in each iteration.

4. Discussion

Why bilinear is not as efficient and stable as negative-exponential kernel?

All the works [2, 10, 32] focusing on irregular kernel consistently adopt the bilinear kernel to interpolate the discrete space. However, we deem it is not good for the greedy searching process.

To better understand the difference, we visualize the kernel shapes and corresponding gradient fields of *bilinear* and *negative-exponential function* in Fig. 3. It is clear that the direction of the bilinear gradient field does not point to the optima on the most coordinates with a significant bias. This can be easily demonstrated by the derivative of bilinear $\mathcal{G}_B(\mathbf{s_y}, \mathbf{y})$. We only show the situation when $\mathbf{y}^x \geqslant \mathbf{s}_\mathbf{y}^x$ and $\mathbf{y}^y \geqslant \mathbf{s}_\mathbf{y}^y$ and the other situations are symmetric:

$$\frac{\partial \mathcal{G}_B}{\partial \mathbf{y}} = trans(\mathbf{y} - \mathbf{s_y}) - \mathbf{1}, \mathbf{y} \geqslant \mathbf{s_y} \tag{17}$$

where $trans(\cdot)$ indicates swapping the x and y coordinates. Obviously Eq. 17 does not point to the same direction as $\mathbf{s_y} - \mathbf{y}$ does. Even worse, bilinear space produces a large biased gradient on the optima $\mathbf{y} = \mathbf{s_y}$, which leads to unstable searching process.

Difference with DCN [2, 32] and ACU [10]

While the motivation of DCN, ACU and the proposed DKE-conv are different, in a specific point of view, all of these try to learn a flexible 'kernel shape' for a convolution layer. The difference between these works can be summarized as follows:

+ **Scoping:** DCN proposes the offset (vector) should differ in each location (context), but the same offset is shared over all input and output channels, so the designed size of offset is $(W, H, 1, 1, 3, 3)$ for (*input width, input height, input channel, output channel, kernel width, kernel height*). ACU proposes to learn an offset shared over pixels and channels, so its shape is $(1, 1, 1, 1, 3, 3)$. In DKE, we do not design but deduce the form of offset from some basic constraints, and demonstrate the offset in the most generalized situation should be shared over spatial location but may differ over channels. Namely, the layer learned by DKE has $(1, 1, K, 1, k')$ offsets. Sec. 5 shows the superiority of this design;

+ **Modality:** DCN actually does not learn the offset directly, its offset is generated dynamically, *i.e.* a linear transform of the local data, similar with STN [8], while DKE and ACU learns a fixed kernel shape and the offset is a part of the network itself;

+ **Learning method:** Both DCN and ACU interpolate the coordinate space by bilinear and learn the parameters/offsets by SGD. DKE evolute the kernel shape by greedy search, which is further embedded in the back-propagation thanks to the novel design of the new interpolative function.

5. DKE in Hard-core Vision Tasks

In this section, we evaluate the DKE on four challenging tasks and various network backbones. All comparable experiments share the same hyper-parameter, such as learning rate, weight decay, momentum, max iteration, etc., and we show the comparison over various backbones, DKE+backbones, and some mentioned related works.

5.1. Close-set Classification

Cifar-10 [13], Cifar-100 [13] and ImageNet [3] are the three most popular object classification datasets, containing 10, 100 and 1000 classes of objects respectively. The ImageNet is much more challenging than the former two and is usually used to evaluate the performance of a new designed neural network architecture. We follow the standard protocol that training and evaluating on the official split and report the top-1 and top-5 error rates.

To evaluate the performance of DKE on shallow and deep neural network, AlexNet [14] and ResNet [6] are adapted to be the backbones. All the convolutional layers in AlexNet are replaced by ACU/DCN/DKE layers. However, since DCN [2] requires heavy workload when the spatial size of input feature map is large, only the last 12 and 30 convolutional layers are replaced in ResNet-18 and ResNet-101. The results are shown in Tab. 1. DKE based backbones outperform baselines consistently and even better than the ACU [10] and DCN [2], which show the superiority of the proposed generalized convolution and searching strategy.

5.2. Open-set Fine-grained Recognition

We adopt Megaface [12] as the fine-grained recognition benchmark in this paper. Megaface is much more challenging than the classification tasks since the distractive class number (1 million) is much higher and the classes in the test set are strictly separated from classes in the training set, so it is a good way to investigate the robustness of neural network architecture.

As the most popular setting in the area of face recognition [4, 22, 25, 24], MS-Celeb-1M [5] is selected as our training data. We follow the same data list, network backbones and loss function in ArcFace [4]. The faces are detected and aligned by RSA [23], and the central 110×110 pixels are cropped and resized to 112 pixels to be the input.

The results are shown in Tab. 2. Besides the consistent gain of DKE, it is interesting that the performances of DCN [2] are even weaker than that of the base model. This is because all the face images are well aligned, but DCN still predicts different offsets for different faces even though on the same location, which introduces uncertainty and noise to the face representation. And a little bit noise on feature would influence the matching result on this 1 *v.s.* 1 million task. This result demonstrates the generalization and robustness of DKE.

5.3. Binary-Class Face detection

Face detection is a binary-class detection task. It is challenging due to the large scale and pose variance in different cases. We follow all hyper parameter settings in STN [1] and the pipeline in RSA [23]. Despite the complex head design in STN, we only adopt the shadow backbone and a simple detection head with a single anchor to be our baseline as in [23, 29], denoted as 'RPN++'. The training data is the same as RSA [23].

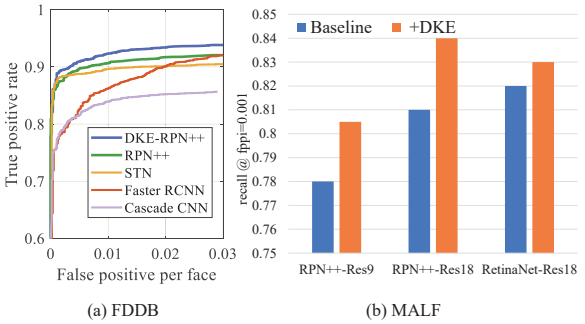
Fig. 4 shows the result on the two famous face detection benchmarks. Since different algorithms are trained by different data and backbones, here we list some related

Model	Depth		Flops	Param.	Cifar-X top-1 err. (%)		ImageNet top-k err. (%)	
	Basic	Special	(GFlops)	(M)	C10	C100	Top-1	Top-5
AlexNet [14]	8	0	0.78	61.1	22.81	56.30	42.8	20.1
ACU-AlexNet[10]	3	5	0.78	61.1	-	-	42.2	20.0
DCN-AlexNet[2]	3	5	0.90	61.4	-	-	40.7	19.3
DKE-AlexNet	3	5	0.82	61.2	**21.4**	**53.1**	**39.1**	**18.2**
ResNet-18	18	0	1.83	11.7	6.8	24.6	30.8	10.9
DKE-ResNet-18	6	12	1.88	11.9	**6.5**	**22.9**	**27.15**	**9.1**
ResNet-101	101	0	7.9	44.5	5.3	21.7	22.7	6.4
ACU-ResNet-101[10]	71	30	7.9	44.5	-	-	21.9	6.1
DCN-ResNet-101[2]	71	30	8.3	46.5	-	-	21.6	5.9
DKE-ResNet-101	71	30	8.0	44.6	**4.7**	**19.9**	**21.1**	**5.7**

Table 1. Accuracy on **Cifar** and **ImageNet** under various neural network architectures. We compare DKE-based networks with their original forms, together with some related works. The 'Basic' and 'Special' under the 'Depth' indicate the number of original and proposed (ACU/DCN/DKE) layers respectively. All the experiments are conducted under the same environment and hyper-parameter configuration. The results are the average of three tries with independent sample of random initialization.

Model	Depth		Top-1 Acc.
	Basic	Special	(%)
R-ResNet-18[4]	18	0	89.4
ACU-R-ResNet-18[10]	6	12	not converge
DCN-R-ResNet-18[2]	6	12	69.5
DKE-ResNet-18	6	12	**91.5**
R-ResNet-101[4]	101	0	97.8
ACU-R-ResNet-101[10]	71	30	not converge
DCN-R-ResNet-101[2]	71	30	97.5
DKE-R-ResNet-101	71	30	**98.0**

Table 2. Top-1 accuracy over the 1 *vs.* 1 million face retrieval benchmark MegaFace. Note that all faces are strictly aligned. ACU fails to converge in both settings, while DCN shows a great level of performance degradation due to the uncertainty it introduces. The result is under the metric of recall ratio at 0.001 false positive per image.



(a) FDDB (b) MALF

Figure 4. Ablative study and comparison with related works on two face detection benchmarks. (a) The precision-recall curve on FDDB benchmarks. Note that we only compare with similar architecture and workload. (b) The comparison on MALF between X and X+DKE, where X denotes different architectures or detection frameworks.

state-of-the-art methods [15, 11] for reference. This results demonstrates that DKE still has the great ability of generalization and reach state-of-the-art performance.

5.4. Multi-Class Object Detection

Model	Backbone	mAP[.5:.95]	mAP[.5]
Faster-RCNN[27]	Res18	28.0	47.3
Faster-RCNN	DKE-Res18	**29.6**	**49.3**
Faster-RCNN	Res50	34.6	55.4
Faster-RCNN	DKE-Res50	**35.3**	**55.9**
FPN[17]	Res50	36.3	58.3
FPN	DKE-Res50	**36.7**	**59.0**

Table 3. Comparison between DKE and basic model on different detection frameworks and backbones.

DKE is also evaluated on MS COCO 2017 [19], the most commonly used object detection dataset. It contains ∼ 120K images of 80 classes. Latest state-of-the-art pipelines FPN [17] and Faster-RCNN [27] are adopted as the Based on the standard evaluation procedure, the original training split and 5,000 miniVal split are used for training and evaluation respectively. Tab. 3 shows the comparison results. Different from the former tasks, DKE slightly improves the performances on different settings. This may because limited extra information and capacity of the network can be achieved by a fixed offset. Nevertheless, the consistent gain still proves the generalization of DKE. Further exploration and research can focus on this task.

6. Investigating on Receptive Field

The kernel coordinate **y** learned by DKE has a high possibility to fall in a non-integer location, and theoretically,

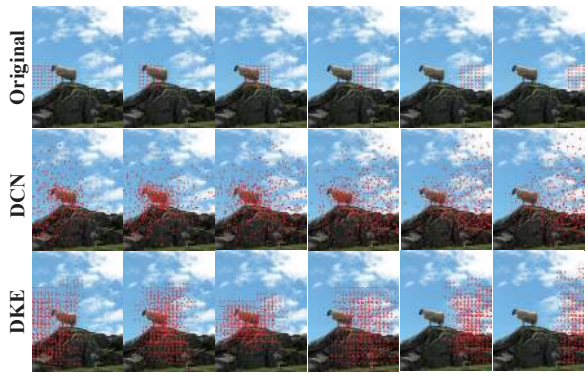


Figure 5. Visualization of sampling kernel shapes. Each image shows $9^3 = 729$ kernel shapes in 3 Res-blocks. In the last row, the most outstanding kernel among the whole input channels (with highest $\mathbf{a}$) is selected to visualize. Note that for the sake of equal comparison, {[-1,-1],[-1,0],...,[1,1]} is used to initialize the $\mathbf{y}$.

the receptive field of this situation is larger than that of the original kernel. Fig. 5 visualizes the receptive field of the original network, DCN and DKE on one pixel of the last feature map of a ResNet-18 network. The receptive field of DKE is indeed larger than the original network.

One may concern the size of the receptive field will influence the performance. We investigate this by enlarging the kernel size and dilation size of layers in baseline model. The same set of layers is modified as DKE. Tab. 4 shows the comparison results on ImageNet. Increasing the receptive field actually leads to better performance. However, the performance of DKE is still ahead by a considerable margin.

Model	Modify	Top-1 err.	Top-5 err.
ResNet-6	original	51.7	26.4
ResNet-6	5×5	46.6	23.1
ResNet-6	dilation 2	48.3	24.7
ResNet-6	dilation 4	47.9	24.1
ResNet-6	DKE	**42.9**	**20.7**
ResNet-18	original	30.8	10.9
ResNet-18	5×5	29.6	10.4
ResNet-18	dilation 2	30.4	10.7
ResNet-18	dilation 4	30.1	10.5
ResNet-18	dilation 6	30.2	10.5
ResNet-18	DKE	**27.15**	**9.1**

Table 4. Receptive field investigation on ImageNet. '5×5', 'dilatoin X' and 'DKE' indicate replacing the same layers from 3×3 `conv` with 5×5 `conv`, 3×3 `conv` with X dilation and DKE, respectively.

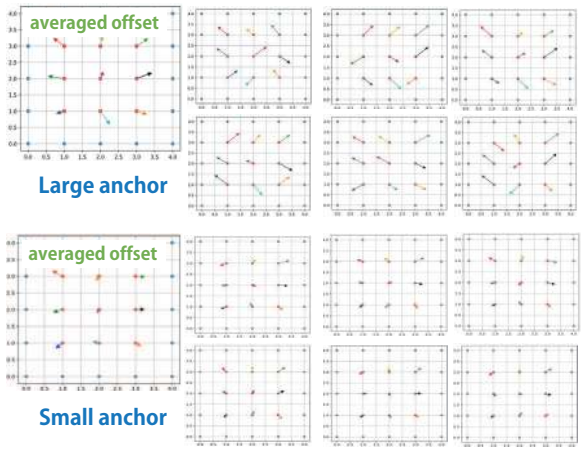


Figure 6. Learned kernel shapes in different branches of a 9-layer face detection network. **Left:** offsets averaged over channels. **Right:** offsets randomly sampled over channels. **Top/bottom:** offsets in the large-anchor/small-anchor branch.

7. Auto-dilated Shallow Network

Recently more and more works [16, 28, 21, 17] are keen on designing multi-branch detector heads for multi-scale object detection. Specifically, the convolution operator of different branches can hold different dilated sizes, and objects with different scales are distributed to different branches.

Inspired by Sec. 6, we think it is interesting to see what if the dilated size of different branches is initialized to 1 and leave the convolution kernel shape to be evolved by DKE.

We use a 9-layer with theoretical receptive field 78×78 to be the backbone of a face detector. The detector split into 2 branches at the 7th layer, and we assign the faces larger than 128 pixels to be the positive samples of the first branch by setting large anchor sizes $(128\sqrt{2}, 256\sqrt{2})$ for it. And for the second branch, we assign three small anchors $(16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2})$ to it.

Fig. 6 shows that DKE tends to expand kernel shapes to acquire more receptive field in the large anchor branch, while slight and irregular offsets are learned in the small anchor branch.

8. Conclusion and Future Work

This work deduces a generalized form for the convolution layer, and a novel greedy searching algorithm is proposed to evolve it from random initialization. The searching algorithm can be embedded into backpropagation thanks to the theoretical research on the interpolative function. Experiments on multiple backbones and tasks show the advancement of this work. Future work should be target on understanding the philosophy behind the learned offset.

References

[1] Dong Chen, Gang Hua, Fang Wen, and Jian Sun. Supervised transformer network for efficient face detection. In *European Conference on Computer Vision*, pages 122–138. Springer, 2016. 6

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 1, 2, 3, 5, 6, 7

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 6

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018. 6, 7

[5] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016. 6

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[7] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990. 1

[8] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2, 6

[9] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, UMass Amherst Technical Report, 2010. 2

[10] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4201–4209, 2017. 1, 2, 5, 6, 7

[11] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster r-cnn. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 650–657. IEEE, 2017. 7

[12] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 2, 6

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2, 6

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 6, 7

[15] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334, 2015. 7

[16] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019. 8

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 2, 7, 8

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 7

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 2

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 8

[22] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*, 2017. 6

[23] Yu Liu, Hongyang Li, Junjie Yan, Fangyin Wei, Xiaogang Wang, and Xiaoou Tang. Recurrent scale approximation for object detection in cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 571–579, 2017. 6

[24] Yu Liu, Guanglu Song, Jing Shao, Xiao Jin, and Xiaogang Wang. Transductive centroid projection for semi-supervised large-scale recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–86, 2018. 6

[25] Yu Liu, Fangyin Wei, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Exploring disentangled feature representation beyond face identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2080–2089, 2018. 6

[26] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2902–2911. JMLR. org, 2017. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7

[28] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 3578–3587, 2018. 8

[29] Guanglu Song, Yu Liu, Ming Jiang, Yujie Wang, Junjie Yan, and Biao Leng. Beyond trade-off: Accelerate fcn-based face detector with higher accuracy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7756–7764, 2018. 6

[30] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. *arXiv preprint arXiv:1807.11626*, 2018. 2

[31] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016. 2

[32] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018. 1, 2, 5, 6

[33] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 2

[34] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 2